REMARKS

SPECIFIC REFERENCE TO EARLIER FILED APPLICATION(S)

As an application in which the benefits of an earlier application are desired must contain a specific reference to the earlier filed application(s) in the first sentence of the specification (37 CFR 1.78), the specification has herein been amended to incorporate such specific reference to earlier filed application(s).

INFORMATION DISCLOSURE STATEMENT AND FORM PTO-1449 LISTING REFERENCE(S) CITED IN ANCESTOR APPLICATION(S)

Submitted herewith under separate cover letter is an Information Disclosure Statement together with Form(s) PTO-1449 listing reference(s) of record in the ancestor application(s) for Examiner initialing to make such art of record in the present application.

CLAIM FOR PRIORITY

Applicant respectfully acknowledges that in order for a patent issuing on the instant application to obtain the benefit of priority under 35 USC 119(a-d) based on priority papers filed in an ancestor application, a claim for such foreign priority must be made in this application. Applicant herein (under separate cover letter) makes such claim for foreign priority, and respectfully submits that the priority papers were filed in ancestor application 09/299,644. Acknowledgment and confirmation of the perfection of Applicants claim for foreign priority are respectfully requested.

Preliminary to the examination of the above-identified application, Applicant herein amends claims 1, 2, 3, 4, 6, 7 and 10. It is respectfully requested that examination be performed on the aforementioned claims.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attachment is captioned **"Version with markings to show changes made."**

To the extent necessary, Applicant petitions for an extension of time under 37 CFR §1.136. Please charge any shortage in the fees due in connection with the filing of this paper, including extension of time fees and excess claim fees, to Deposit Account No. 01-2135 (referencing case No. 500.37167CX1) and please credit any excess fees to such deposit account.

Respectfully submitted,

Frederick D. Bailey
Registration No. 42,282
ANTONELLI, TERRY, STOUT & KRAUS, LLP

FDB/pay
(703) 312-6600

IN THE SPECIFICATION

Please amend the specification as follows:

Page 1, between lines 1 and 2, insert --This is a continuation of prior application Serial No. 09/299,644, filed April 27, 1999.--

Please replace the paragraph beginning at page 7, line 25, with the following rewritten paragraph:

--In Fig. 2, a client program 111 has therein: program data 113 and 122 which is specific to a computer and a described program language; an ORB 141 which is used by the client program; and the like. The OBR 141 includes a marshalling part 114, a unmarshalling part 121, cache registering part 151 and a cache part 131. A server program 112 has therein: program data 117 and 118 which is specific to the computer and the described program language; and ORB 142 which is used by the server program; and the like. The OBR 142 includes a marshalling part 119, a unmarshalling part 116, cache registering part 152 and a cache part 132.--

Please replace the paragraph beginning at page 8, line 6, with the following rewritten paragraph:

--How to exchange the data between the client program 111 and server program 112 in Fig. 2 will now be described with respect to a flow of data. In the client program 11, to transmit the program data 113 of a data format specific to the computer on which the client program operates and the described program to the

server program 112, a marshalling part 114 of the ORB 141 converts [from] the program data 113 [to request] requesting communication data 115 which is not specific to particular computers or program languages and the cache registering part 151 registers the result of the conversion into the cache 131. In this instance, if all or a part of the program data 113 exists in a cache 131 of the client program by referring to the cache 131, the data is converted into the request communication data 115 by using the cache data. After completion of the conversion, the data 115 is transmitted to the server program 112.--

Please replace the paragraph beginning at page 8, line 22, with the following rewritten paragraph:

--In the server program 112, an unmarshalling part 116 of the ORB 142 converts the received request communication data 115 into the program data 117 of a format which is specific to the computer on which the server program operates or to the described program language and the cache registering part 152 registers the result of the conversion into the cache 132. In this instance, if all or a part of the request communication data 115 exists in a cache 132 of the server program by referring to the cache 132, the data is converted into the program data 117 by using the cache data.--

Please replace the paragraph beginning at page 9, line 4, with the following rewritten paragraph:

--After the server program processed the request from the client, to return the program data 118 as a processing result to the client program, the data is converted

15

into response communication data 120 by a marshalling part 119 of the ORB 142 and the cache registering part 152 registers the result of the conversion into the cache 132. At this time, if all or a part of the program data 118 exists in the cache 132 of the server program with reference to the cache 132, the data is converted into the response communication data 120 by using the cache data. The response communication data 120 is transmitted to the client program.--

Please replace the paragraph beginning at page 9, line 15, with the following rewritten paragraph:

--In the client program 111, the received data is converted into the program data 122 of the client program language-specific format by an unmarshalling part 121 of the ORB 141 and the cache registering part 151 registers the result of the conversion into the cache 131. At this time, if all or a part of the response communication data 120 exists in a cache 131, the data is converted into the program data 122 by using the cache data. In this manner, the data transmission and reception can be performed between the distributed client and server programs at a high speed.--

IN THE CLAIMS

1. (Amended) An apparatus for converting data of a first data format used on a particular computer into data of a second data format which is common to a plurality of computers in a distributed computing environment, comprising:

a caching part for storing [a correspondence between data of the first data format and corresponding data of the second data format] data of the first data

16

format and corresponding data of the second data format and correspondence between the data of the first data format and the corresponding data of the second data format;

a marshalling part for discriminating whether at least a part of the data of the first data format to be converted has been stored in said caching part or not, for converting said data of the first data format to be converted into the data of the second format by using the [data on] correspondence stored in said caching part such that; for a part or whole of said data of the first format to be converted that matches the data of the first format stored in said caching part, conversion is made by using the correspondence and that; for another part of said data of the first format to be converted that does not match the data of the first format stored in said caching part, conversion is made with or without using said caching part depending upon a data type of said unmatched part of the data of the first data format [when said data exists, and for converting said data to be converted into the data of the second data format without using said caching part when said data does not exist]; and

a cache registering part for registering said converted data into said caching part.

2. (Amended) An apparatus for converting data of a second data format which is common to a plurality of computers into data of a [second] first data format used on a particular computer in a distributed computing environment, comprising:

a caching part for storing [a correspondence between the data of the second data format and the data of the first data format] data of the second format and

corresponding data of the first format and correspondence between the data of the second format and the corresponding data of the first format;

an unmarshalling part for discriminating whether at least a part of the data of the second data format to be converted has been stored in said caching part or not, for converting said data of the second data format to be converted into the data of the first data format by using the [data on] correspondence stored in said caching part such that; for a part or whole of said data of the second format to be converted that matches the data of the second format stored in said caching part, conversion is made by using the correspondence and that; for another part of said data of the second format to be converted that does not match the data of the second format stored in said caching part, conversion is made with or without using said caching part depending upon a data type of said unmatched part of the data of the second data format [when said data exists, and for converting said data of the second data format to be converted into the data of the first data format without using said caching part when said data does not exist]; and

a cache registering part for registering said converted data of the first data format into said caching part.

3. (Amended) A method of converting client program data of a client program into request communication data common to a plurality of program languages or computers in order to request a processing task from said client program to a server program on a server in a distributed computing environment, comprising the steps of:

discriminating whether at least a part of said client program data exists [on] in

18

a cache of said client program or not by referring to said cache, said cache storing client program data and corresponding request communication data and correspondence between the client program data and the request communication data;

converting said client program data into said request communication data by using the [program data existing on] correspondence stored in said cache such that; for a part or whole of the client program data to be converted that matches the client program data stored in said cache, conversion is made by using the correspondence and that; for another part of said data of the client program data to be converted that does not match the client program data stored in said cache, conversion is made with or without using said cache depending upon a data type of said unmatched part of the data of the client program data [when said data exists and converting said data based on attributes of said client program data when said data does not exist];

storing the request communication data into said cache as a conversion result when said data does not exist; and

transmitting said converted requesting communication data to said server program.

4. (Amended) A method of converting request communication data which is transmitted from a client and is common to a plurality of program languages or computers into program data of a program language describing a server program which operates on a server in a distributed computing environment and processing a processing task requested by the client, comprising the steps of:

receiving said request communication data by an unmarshalling part in said

server;

discriminating whether at least a part of said request communication data exists in a cache of said server program or not by referring to said cache, said cache storing the request communication data and corresponding program data and correspondence between the request communication data and the program data;

converting said received requesting communication data into the program data of the program language describing said server program by using the [request communication data existing in] correspondence stored in said cache such that; for a part or whole of the request communication data to be converted that matches the request communication data stored in said cache, conversion is made by using the correspondence and that; for another part of said data of the request communication data to be converted that does not match the request communication data stored in said cache, conversion is made with or without using said cache depending upon a data type of said unmatched part of the data of the request communication data [when said data exists in said cache and converting said data based on attributes of said received requesting communication data when said data does not exist];

storing the program data into said cache as a conversion result when said data does not exist; and

processing the processing task requested by said client on the basis of said converted program data.

6. (Amended) A method whereby a client program receives and processes response communication data which is transmitted from a server and is common to a plurality of computers or program languages in a distributed computing

20

environment, comprising the steps of:

receiving said response communication data by said client program;

discriminating whether at least a part of said response communication data exists [on] in a cache of said client program or not by referring to said cache, said cache storing response communication data and corresponding client program data and correspondence between the response communication data and the client program data;

converting said received response communication data into client program data by using the [communication data on] correspondence stored in said cache such that; for a part or whole of the response communication data to be converted that matches the response communication data stored in said cache, conversion is made by using the correspondence and that; for another part of said data of the response communication data to be converted that does not match the response communication data stored in said cache, conversion is made with or without using said cache depending upon a data type of said unmatched part of the data of the response communication data [when said data exists and converting said data by discriminating attributes of said received response communication data when said data does not exist]; and

storing the client program data into said cache as a conversion result when said data does not exist.

7. (Amended) A computer software product including a computer-readable medium having a computer readable program embodied in the medium for making program data to be transmitted subject to a marshalling process to generate request

communication data, said medium having stored thereon:

a program code portion for comparing said program data to be transmitted with [said] program data on a cache, said cache storing a pair of contents of said program data and contents of [the] communication data corresponding thereto for every type of the program data;

a program code portion for, when the contents of said program data to be transmitted and the contents of the program data on said cache are matched, comparing subsequent program data to be transmitted with said program data on said cache until a difference is detected;

a program code portion for copying the communication data on said cache corresponding to the program data which is matched until said difference is detected into [requesting] request communication data;

a program code portion for, when a difference is detected between contents of certain program data to be transmitted and contents of the program data on the cache, forming corresponding communication data from the program data in which said difference is detected in accordance with a type of the program data in which said difference is detected;

a program code portion for storing a correspondence between the program [code] data in which said difference is detected and said formed communication data onto said cache; and

a program code portion for, when there is no transmission program data to be compared, copying the formed communication data remaining on said cache into said request communication data.

22

10. (Amended) A computer software product including a computer-readable medium having a computer readable program embodied in said medium for making received request communication data subject to an unmarshalling process to generate output program data, said medium having stored thereon:

a program code portion for comparing said received request communication data with [said] communication data on a cache in units of a predetermined communication data length, said cache storing a pair of contents of said [program] received request communication data and contents of the communication data corresponding thereto for every type of the [program] received request communication data;

a program code portion for, when the contents of said received request communication data and the contents of the communication data on said cache are identical, comparing subsequent received request communication data with said communication data on said cache until a difference is detected;

a program code portion for copying the communication data on said cache corresponding to the received request communication data which is matched until said difference is detected into said output program data;

a program code portion for, when a difference is detected between contents of certain received request communication data and contents of the communication data on the cache, forming corresponding program data from said received request communication data in which said difference is detected in accordance with a type of said received request communication data;

a program code portion for storing a correspondence between said received request communication data in which said difference is detected and said formed

corresponding program data onto said cache; and

a program code portion for, when there is no received request communication data to be compared, copying the formed corresponding program data remaining on said cache into said output program data.